

- 1. LanguageDefinition.lblProgram
- 2. ProgramNo
- 3. ProgramName
- 4. LanguageDefinition.lblStep
- 5. StepNo
- 6. StepName
- 7. LanguageDefinition.lblActual
- 8. LanguageDefinition.lblNominal
- 9. MainValueList. MainValue1
- 10. MainValueList. MainValue2
- 11. MainValueList. MainValue3
- 12. MainValueList. MainValue4
- 13. StepList.Header lub RelayList.Header
- 14. StepList.Data lub RelayList.Data
- 15. StatusBarMessage scroll, scroll area starts from statusbar time
- 16. StatusBarTime
- 17. LanguageDefinition.cmdBottom1
- 18. LanguageDefinition.cmdBottom4
- 19. LanguageDefinition.cmdBottom5

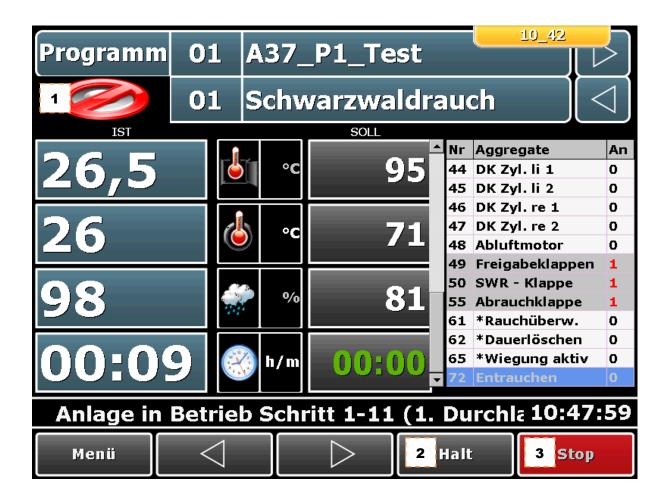
Pole "HtmlTitle" idzie jako <HTML><TITLE>

Kolory listy po prawej stronie - kolor tekstu "GridForeColor", kolor tła "GridBackColor", format:" index", "wartość RGB", np.:

```
"0":"#F5F5F5",
"1":"#FFFF00",
"2":"#6495ED",
"3":"#008000"
```

W danych wygląda to tak - brak informacji o kolorze, używamy kolor z indeksem zero "0", jeśli inaczej, to przesyłany jest nr koloru "color=X", np.:

```
"Data":[
 [
   "02",
   "Umluftmotor",
   "0",
   1111
 ],
   "17",
   "Reinigungspumpe",
   "1",
   "color=1"
 ],
 [
   "20",
   "Rauchklappe",
   "1",
   "color=2"
 ],
   "36",
   "0",
   "color=3"
 ],
```



- 1. ButtonStatus.lblStep = "" or "icon=alarm.png" if empty string display as normal, when icon=alarm.png then place assets/images/alarm.png on black background instead of standard display
- 2. ButtonStatus.cmdBottom4 = "" or "css=gradientRed" or "css=gradientRed,blink"
- 3. ButtonStatus.cmdBottom5 = "" or "css=gradientRed" or "css=gradientRed,blink"
- if empty string display as normal, when "css=gradientRed" then use red button layout, when "blink" than blink html button